Gujarati Character Recognition

Sameer Antani
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802, USA
antani@cse.psu.edu

Lalitha Agnihotri
Philips Research Briarcliff,
Briarcliff Manor, NY 10510, USA
laa@philabs.research.philips.com

Abstract

This paper describes the classification of a subset of printed or digitized Gujarati characters. Gujarati belongs to the genre of Devanagri scripts from the Indian subcontinent. Very little work is found in the literature for recognition of Indian language scripts. For this paper a subset of similar appearing Gujarati characters was chosen and subjected to classification by different classifiers. The sample and test images for the characters were obtained from digital images available on the Internet and from scanned images of printed Gujarati text. For their classification, the Euclidean Minimum Distance and the k–Nearest Neighbor classifiers were used with regular and invariant moments. The characters were also classified in the binary feature space using Hamming Distance classifier. The paper presents the recognition rates for these classifiers. A recognition rate of 67% *is achieved. The work described in this paper is preliminary; however, since ICDAR'99 is being held in India, we hope that this would be of interest to the participants.*

1 Introduction

This paper addresses the issue of character recognition of the Gujarati language from the Indian subcontinent. Gujarati belongs to the genre of languages that use variants of the Devanagri script. No significant work is found in the literature that addresses the recognition of Gujarati language. The Gujarati script, shown in Figure 1, is derived from the Devanagri script. Other languages like Sanskrit, Hindi, Marathi use a similar script. Some of the Gujarati characters are very similar in appearance. With sufficient noise these characters can easily be misclassified. Often, these characters are misclassified even by humans who then need to use context knowledge to correct the error. Unlike Devanagri, the Gujarati characters within a word are are separated by white space. The intra-word characters in the Devanagri are connected with an over stroke. This eases the problem of character separation in a word. This paper addresses the problem of discriminating between subset of such characters. The paper describes the results obtained by using the Euclidean Minimum Distance classifier, the *k*–Nearest Neighbor classifier, and the Hamming Distance classifier. The features used for these classifiers are the regular moments, Hu invariant moments and the distribution in the binary feature space.

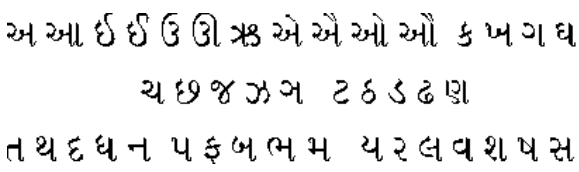

Figure 1. *A subset of the Gujarati Alphabet*

1.1 Previous Work

Very little work is seen in the literature on recognition of Indian languages in general and Gujarati in particular. Work done for recognition of some other Indian scripts such as Tamil, Telugu and Bangla have been described in [8]. A feature based approach has been adopted by [10] for Telugu script recognition which works on isolated characters. The features used pertain to X and Y extrema of the character. Some multi-lingual character recognition work found in the literature is included below. For a language independent and segmentation free technique, a Hidden Markov Model (HMM) based OCR has been proposed by [6]. Several other citations on OCR using HMM are available in [8]. A comprehensive discussion on a complete text reading system is available in [13]. Texts by Nadler and Smith [7], Schallkoff [11], Schürmann [12], Gose et al [5] and Dori [3] provide useful information on classifiers and the design of a character recognition system.

Typically, character recognition work addresses the problems posed by translation, rotation and scaling. Inclusion of features that are invariant to these affine transformations is important and are included in this experiment. The characters currently used in the data set used for classification are of a fixed size. Section 2 illustrates some typical characteristics of Indian languages. Section 3 describes the methods adopted for the generation of the data set in greater detail. The remainder of the paper is as follows. Section 4 describes the features which will be used for classification. Section 5 describes the selected classifiers. Section 6 presents the classification results and makes observations on them. We conclude with Section 7.

2 Characteristics of Gujarati Script

This section illustrates some of the issues which characterize the Gujarati script. Although we discuss the issues with reference to Gujarati, they are largely applicable to most Indian languages. As mentioned before, the characters within a word in Gujarati are separated (as in Latin languages) by white-space. Each of the consonants in Gujarati can have one of 11 strokes adjacent to, over, or under it. These represent the vowels. Ligatures also are very common and can themselves have the vowel strokes around them. An OCR system would thus need to handle a large variety of fonts along with these characteristics. The association of the vowel stroke with the character is in itself an interesting problem for recognition.

3 Generation of Data Set

The data samples used in the experiment were obtained from various sites on the Internet and from scanned images of printed Gujarati text. The characters were then manually cropped from the scanned images and resized to the selected size. The images found on the Internet are pre-digitized, noise- and skew-free images. In contrast the scanned images (scanned at 100 dpi) possibly have skew in them and also may have noise pixels. Some characters were also observed to be broken at locations which have fine links. The images in the database are from 15 font families. The number of unique samples available to us were few. The sample (training) data set consists of 10 characters (classes) each having 10 samples. These have been created from 5 different fonts. The characters have been extracted from images of these fonts. The test set consists of 30 samples of each character.

To address the problem of a small number of sample data available, a larger set was artificially created by duplicating the available character images. To address the second problem of lack of variety in the images, a method suggested in [1] was adopted. The images are scaled up and then scaled down to a fixed size. In the experimental data set, each character is of the same size. Any of the sample images found that were of a different size were scaled to this size. Due to the digitizing effect of scaling, the new scaled character would represent the *real* noise found in typical samples. A need for this scaling is seconded because of the perfect character property in a part of our sample set described above. Other methods of introducing noise described in the literature include *stroke-thinning*, *stroke-thickening*, blurring, skew, width change, height change, kerning [1] etc. Scaling is used to achieve the resolution effect described.

It is important to note that this system does not have the usual preprocessing phases that separate words from sentences and characters from words. It also does not have skew correction or noise removal etc. These are preprocessing phases in a typical text reading system as described in [13]. Since this is preliminary work, little effort was spent on implementing these phases. Figure 3 shows the subset of characters selected for this experiment. The characters have the following phonetics, “sa”, “kha”, “ka”, “fa”, “ya”, “a”, “cha”, “ba”, “ha”, “ja”. We will be using the first letter of the phonetic of each character (except for “kha” and “cha”, for which we use the first two) to represent it in the confusion matrices presented in Section 6.

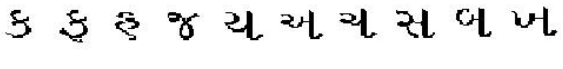

Figure 2. *The selected characters*

4 Feature Selection

Feature selection is one of the most important steps in developing a classification system. This section describes the various features selected by us for classification of the selected characters. In selecting features for a classification system, it is necessary to study the characteristics of the script being classified. This would help in selecting features that better discriminate the character samples.

Since all the characters in our sample set are of the same size, we expect to benefit from the first order and higher order moments. The characters in Figure 3 are very similar. For example, “ka” and “fa” are alike (1^{st} and 2^{nd} characters in the figure), so are “ya”, “a” and “cha” (5^{th}, 6^{th} and 7^{th} characters respectively), and likewise “ba” and “kha” (the 9^{th} and 10^{th} characters). Also, the images of the characters are of the same size, and the characters themselves are approximately the same size. The size and the location of the character within the selected size weren’t controlled.

The classifier uses both the invariant moments and the raw moments for classification. The selected raw moments are M(0,0), M(0,2), M(2,0), M(1,1), M(0,3), M(3,0), M(1,2) and M(2,1). Additional information on moments can be found in [4].

Here $B[.][.]$ represents the image. These moments will be used in the classifiers listed in the next section. Other features used by the classifiers are the image pixel values themselves. The pixels form a feature space of the size of the image. The images in our data set are 30x20 pixels, thus creating a 600 dimensioned binary feature space.

The translation invariant moments are found by normalizing the above stated centralized moments as follows and making a combination of two or more of the normalized centralized moments. The 7 Hu invariant moments used are described in [9].

5 Selected Classifiers

The classifiers are selected based on our experiences with various classifiers studied for a subset of the English alphabet and ideas borrowed from [8]. The classifiers selected are the *k*–Nearest Neighbor and the Euclidean Minimum Distance Classifier using Regular moments in 8 dimensions and Invariant moments in 6 dimensions and in the 600-dimensional binary space. Additionally, the classifier based on the Minimum Hamming Distance in the 600-dimensional binary space was also used to classify the characters. These classifiers are described below.

I **The k-Nearest Neighbor Classifier:** The *k*–Nearest Neighbor classifier [2] has been found to give fairly good results for English characters. We assume the error cost to be equal for each class. The *k*–NN classifier then votes among the k closest samples to a test sample and identifies it to that class which has the largest number of votes. The reader is also referred to Figure 3 which shows the high correlation between different characters. There are many such examples in the English script which the *k*–NN classifier should be able to distinguish well. The nearest neighbor is found by using the Euclidean distance measure. Additional information on the *k*–NN classifier can be found in [5].

II **The Minimum Hamming Distance Classifier:** The Minimum Hamming Distance Classifier uses the Hamming Distance between the sample and the class centroids built using the training sets to classify characters. It is assumed that the image pixels have a Bernoulli distribution. Then the hamming distance is the sum of the absolute pixel difference (in binary space) between the class centroids and the image of the character being classified. The class centroids are either 0 or 1 for each dimension, the value of which is determined by majority for that pixel location in the training set.

6 Observations

The classification results are presented below in confusion matrices. A confusion matrix compares the classes and classification results. Each row is assigned to one class and each column is assigned a character. All locations are initialized to 0. The count is incremented at the location for which a particular character is classified. The last row and column (E1 and E2) count the number of misclassified elements, indicating the number of errors for each class and character. The location common to row E1 and column E2 indicates the total error.

As shown in Table 1, the best recognition rate for was 67%, obtained from the 1-NN classifier in the 600 dimension binary feature space. As shown in Table 2, 1-NN classifier in the regular moment space resulted in a recognition rate of 48%. The *k*–NN classifier with k set to values 2, 3, 4, 6, and 10 resulted in progressively poorer recognition rates. Table 3 details the results from the Minimum Hamming Distance classifier having a recognition rate of 39%. The Euclidean Minimum Distance Classifier which recognized only 41.33% of the test set characters in the regular moment space. The invariant moment features were found to have low recognition rates. The highest recognition rate was 29% for the 1-NN classifier for k set to 4. With the Euclidean Minimum Distance recognizing only 23% of the test characters.

From the results in Table 1 and concentrating on column E1 we see that “cha” and “kha” have high misclassification. While “kha” has been largely misclassified as “ba” (as expected), the misclassification error for “cha” is spread over several characters. Some of these have high shape correlation with “cha”. From Table 2 for 1-NN with regular moments, we observe that the recognition rate isn’t as impressive. Yet, no single character was misclassified as a single other character. The errors are spread out. A similar observation can be drawn from the results of the Minimum Hamming Distance classifier. The NN–Classifier using moments (regular or invariant) depends on the shape and size of the characters. The characters in our dataset are of different stroke thicknesses, are sometimes broken and suffer from resolution effects. All of these cause such classifiers to perform poorly. The NN–Classifier in the binary feature space performed well because it allowed searches beyond the exact shapes learned from the training samples.

7 Conclusion

The test set used in this experiment was rather small. The toughest phase in the experiment was getting a good set of characters for classification. This highlights the need for generation of a large ground-truthed set of characters of various resolutions so that more research can be performed for recognition of languages from the Indian subcontinent. Also, the characters used for the experiment were enclosed in a bounding region of a fixed size. Although this may not always be the case, attention is drawn towards the fact that mere shape based recognition will not always perform well. Especially when the system is dealing with ligatures. Different font families represent the same character differently and the correlation between similar characters is varies from font to font. This preliminary research helped us focus our attention on these matters so that issues for building a robust character recognition can be studied.

	a	b	ch	f	h	j	k	kh	s	y	E1
a	22	3	2	0	0	0	0	2	0	1	8
b	2	25	0	0	0	0	0	1	1	1	5
c	5	4	7	0	0	0	0	1	7	6	23
f	0	0	0	27	0	0	3	0	0	0	3
h	2	0	0	2	20	1	1	0	1	3	10
j	0	1	0	0	1	27	0	1	0	0	3
k	0	1	0	1	1	0	27	0	0	0	3
kh	7	18	0	0	0	1	0	4	0	0	26
s	1	0	3	0	0	0	0	2	22	2	8
y	1	1	0	0	0	2	0	0	6	20	10
E2	18	28	5	3	2	4	4	7	15	13	99

Table 1. 1-NN - Binary Feature Space

	a	b	ch	f	h	j	k	kh	s	y	E1
a	13	1	6	0	0	0	0	1	4	5	17
b	3	8	8	0	0	0	0	3	0	8	22
c	1	7	10	0	1	0	0	1	1	9	20
f	1	1	1	20	1	0	4	1	1	0	10
h	1	0	0	2	20	0	2	0	2	3	10
j	0	0	0	1	0	26	0	3	0	0	4
k	0	0	0	8	0	0	21	0	1	0	9
kh	2	8	0	0	0	3	0	8	3	6	22
s	8	0	4	5	0	0	2	3	7	1	23
y	6	5	4	0	0	0	0	3	1	11	19
E2	22	22	23	16	2	3	8	15	13	32	156

Table 2. 1-NN - Regular Moments

	a	b	ch	f	h	j	k	kh	s	y	E1
a	19	2	0	1	0	0	4	2	0	2	11
b	5	16	3	0	0	0	6	0	0	0	14
c	10	6	5	0	0	0	4	0	4	1	25
f	1	4	0	16	2	1	3	0	3	0	14
h	1	1	0	10	9	0	3	0	3	3	21
j	4	3	0	1	3	15	4	0	0	0	15
k	0	9	0	1	0	0	16	1	2	1	14
kh	17	6	0	0	0	0	2	4	0	1	26
s	5	2	5	2	0	0	3	2	10	1	20
y	2	0	2	5	4	0	3	2	5	7	23
E2	45	33	10	20	9	1	32	7	17	9	183

Table 3. Min. Hamming Distance Classifier

References

[1] H. S. Baird. Document Image Defect Models. In L. O'Gorman and R. Kasturi, editors, *Document Image Analysis*, pages 315–325, 1995.

[2] B. V. Dasarathy. *Nearest neighbor (NN) norms, NN pattern classification techniques.* 1991.

[3] D. Dori and A. BruckStein. *Shape, Structure and Pattern Recognition.* 1994.

[4] R. C. Gonzales and R. E. Woods. *Digital Image Processing.* Addison Wesley, 1992.

[5] E. Gose, R. Johnsonbaugh, and S. Jost. *Pattern Recognition and Image Analysis.* Prentice-Hall, 1996.

[6] J. Makhoul, R. Schwartz, C. LaPre, C. Raphael, and I. Bazzi. Language independent and segmentation free technique for OCR. In *IAPR Workshop on Document Analysis Systems*, pages 99–115, 1996.

[7] M. Nadler and E. P. Smith. *Pattern Recognition Engineering.* 1993.

[8] U. Pal. *On Development of an OCR System for Printed Bangla Script.* PhD thesis, Computer Vision and Pattern Recognition Unit, ISI, Calcutta, India, 1997.

[9] W. K. Pratt. *Digital Image Processing.* Wiley Interscience, 1991.

[10] P. Rao and T. Ajitha. Telugu script recognition. In *International Conference on Document Analysis and Recognition*, pages 323–326, 1995.

[11] R. Schallkoff. *Pattern Recognition - Statistical, Structural and Neural Approaches.* 1992.

[12] J. Schürmann. *Pattern Classification - A Unified View of Statistical and Neural Approaches.* John Wiley & Sons, Inc., 1996.

[13] S. Tsujimoto and H. Asada. Major component of a complete text reading system. In L. O'Gorman and R. Kasturi, editors, *Document Image Analysis*, pages 298–314, 1995.